

CALIFORNIA POLYTECHNIC UNIVERSITY SAN LUIS OBISPO

CSC SENIOR PROJECT

FORMAL REPORT

Ciceroni iOS App

Abstract

The Ciceroni app for iOS is a crowdsourcing tour app that allows users to create and share tours consisting of a collection of locations. Although it can be used for various purposes, it was designed to facilitate the transfer of knowledge from locals to tourists and to provide a free alternative to traditional tours. The app enables Facebook users to create and share lists of locations which other users can then view and get directions to on a map. This project was developed in HTML 5 and JavaScript and demonstrates the use of the Google Maps JavaScript API, Parse Core database, and several Cordova plugins.

Author:

Forrest D. HANSEN

Supervisor:

Zoë J. WOOD

June 12, 2015

Contents

1	Problem Description	2
2	Relevant Works	3
2.1	Previous Work	3
2.2	Related Work	3
3	Problem Solution	4
3.1	Overview	4
3.2	Technical Accomplishments	4
4	Results	6
4.1	Creating a ciceroni	6
4.2	Viewing ciceroni	10
5	Intel XDK Review	14
5.1	Advantages	14
5.2	Disadvantages	14
6	Future Work	15
7	Conclusions	15
7.1	Limitations	15
7.2	Retrospective	16

1 Problem Description

Tourists visiting new places are often overwhelmed by the number of attractions. Some people prefer to explore and discover places on their own. People that don't want to miss out on the best attractions inevitably turn to the internet for help in guiding them. This approach may or may not lead to finding helpful information depending on the location the tourists find themselves in. This is because many lesser visited locations may have little to no information about what places are the best for tourists to visit. Locals to the area will obviously have knowledge about where they should go, so the problem is how to transfer the knowledge of the locals to the tourists. Even some major tourist destinations with a plethora of tour guide information may be missing some well-kept secret places that only locals know about. Tourists need some avenue of information transfer to learn about the best places to visit from the people who know the most about the area.

Some popular tourist attractions employ tour guides to provide tourists with a directed and informational trip. While this is a popular approach, there are a couple of reasons for tourists to be unhappy with these tours. People are generally happier going at their own pace, but tour guides force tourists to stay on a set path and prevent them from exploring at their leisure. In some cases, tours of this nature prevent tourists from taking breaks in order to keep the tour going. Another reason for tourists to avoid these types of guided tours is because they cost money. While on vacation, many tourists will inevitably end up spending lots of money and they will appreciate any chance to save some cash, but they also don't want to miss out on any interesting information about each place of interest. Tourists need some method of learning about an area without spending money and without being locked into a fixed tour.

The Ciceroni app aims to solve both these problems by providing free self-guided tours created by users themselves. This enables anybody with good tourism tips of an area to share their knowledge. The tours are simply a list of locations which can be viewed on a map. The tours are self-guided because users choose the order in which they visit locations and are free to move at their own pace as well as explore other oddities which catch their eye along the way.

2 Relevant Works

2.1 Previous Work

There are a plethora of virtual tours available online that allow anyone with an internet connection to virtually visit a place of interest. These types of applications are not relevant to the problem domain because they are not designed to be used by tourists who are physically traveling to the places of interest.

Many other applications or online tour guides provide tours of specific areas. For example, the Stray Boots app available on the Apple App Store provides self guided tours of over 60 locations [1]. However, by not allowing users to create and share their own tours with the general public, the app doesn't solve the first problem mentioned in Section 1 about transferring the knowledge of locals to tourists. Of course some of the preset tours available from these types of apps could be created by locals and provide tourists with the best possible experience. However, the fact that normal users cannot create their own tours or add information to existing tours means that there is a high probability that some great tourist attractions or interesting information will be missing.

The app that comes the closest to providing user-created and self-guided tours is the TrackMyTour app available from the Apple App Store [2]. The app allows users to create tours comprised of several waypoints, which can then be made publicly available. It doesn't fully solve the problem of providing self-guided tours because it doesn't provide any way for users to get directions to a specific waypoint. Each waypoint simply has a title, description, and optional photo. All the waypoints in a tour can be viewed on a map, but there is no option to get directions and no address to use in other mapping software.

2.2 Related Work

Cordova plugins provide functionality for native HTML5 web apps. Several of these plugins are used in the Ciceroni app. The Cordova Camera plugin allows native web apps to access the camera on a mobile device [3]. The Cordova Media plugin allows sound effects or music to be played in native web apps [6]. The Cordova Geolocation plugin provides Javascript methods for accessing a mobile device's current GPS location from within a native web app [4]. The Cordova

In-App Web Browser plugin provides simple web browser functionality entirely contained within a native web app [5]. The Facebook Connect plugin is used to provide login functionality using a Facebook account [7].

3 Problem Solution

3.1 Overview

The Ciceroni iOS app was designed to solve the problems elaborated upon in Section 1. It allows any user with a Facebook account to create tours nicknamed ciceroni which are comprised of a set of locations. A cicerone has a title, description, and a category and can be set to public or private. Each location can contain a title, description, GPS location, picture, and an internet link. In addition to creating ciceroni, the app allows users to search for public ciceroni created by other users. They can search by keyword and filter by category or they can search by proximity to a chosen location. Opening a cicerone will display the information added by the author. Each location in the cicerone can be opened to view its information or all the locations can be displayed on a map. Directions to a location can be obtained via the map view.

3.2 Technical Accomplishments

The Ciceroni app leverages five Cordova plugins to provide various functionalities. The Camera plugin is used to allow users to capture images of each location [3]. The Media plugin is used to play the sound effects when buttons are clicked [6]. The Geolocation plugin is used to obtain latitude and longitude coordinates for each location as well as obtain the devices current location for proximity searches [4]. The In-App Web Browser plugin allows users to view the internet links associated with some locations entirely within the app [5]. The Facebook Connect plugin is used to provide login functionality using a Facebook account [7]. These plugins were installed by simply checking a checkbox in the project settings within the Intel XDK development environment. The proper use for each plugin was determined by reading the documentation on their respective GitHub pages.

Parse Core is the database service which Ciceroni utilizes to save and query

ciceroni. To use the database service, the Ciceroni app had to be registered with Parse to obtain an Application ID and a JavaScript Key. Once initialized with these keys, the Parse JavaScript API provided the functionality necessary to easily save an entire cicerone's data except for the location images [8]. Data is saved to a Parse database in the form of a Parse Object. However, Parse Objects are limited to a maximum size of 128kB. This limitation forces users of the database to place large image files in separate Parse Files, which have larger maximum size limits. The Ciceroni app converts each location image to a base64 string which is then saved as a Parse File. Each Parse File containing image data is then linked to a Parse Object whose unique object ID is then saved within the cicerone data as a string. Location images are retrieved by querying the database for Parse Objects that match the saved object IDs contained in the cicerone data and then retrieving the linked Parse File containing the image data.

The Facebook Connect plugin was utilized to provide login functionality. This plugin is essentially a wrapper to the Facebook JavaScript API [7]. This plugin required that the app be registered with Facebook in order to obtain a Facebook App ID. Once setup, this plugin allows the app to obtain unique user ID values which are associated with each cicerone the user creates and are saved along with the ciceroni to the database. These user ID values are then used to retrieve a user's previously created ciceroni when they next login. As part of the initialization process, the Ciceroni app checks the login status of the previous user and automatically skips the login step if they are still logged in.

The Ciceroni app uses the Google Maps JavaScript API to provide dynamic maps for selecting and viewing locations [9]. The app imports the necessary JavaScript files and creates a map display within a specified JavaScript div element. The map display is slightly different for each use, whether it be selecting a location or viewing a location. When the user needs to select a GPS location, the map is centered at the device's current location and contains a search box for querying known locations. The current location of the device is obtained using the Cordova Geolocation plugin which is then passed to a Google Maps JavaScript API method. The search box functionality is provided as part of the Google Maps JavaScript API after attaching a JavaScript text input element to the map by using another API method. After centering the map on the desired location, users can select a location by touching anywhere within the map which will cause a map

marker to appear at that location. When the user simply needs to view a location on the map, the map is centered on that location and no search box is provided. When viewing locations, users can click a location, causing its image marker to enlarge. This is accomplished by attaching a click event handler to the image marker upon their creation. When the “Get Directions” button is clicked, another Google Maps JavaScript API method is used to display a path from the device’s current location to the selected location. Detailed step-by-step instructions are also made available by attaching another JavaScript div element to the map via yet another Google Maps JavaScript API method.

4 Results

The final version of the Ciceroni app provides a solution for both problems mentioned in Section 1. Local residents of any tourist destination can easily create a simple tour of their favorite places, thereby making their detailed knowledge available to the public. It provides tourists with a free alternative to traditional tours that can provide the same information but which also allows them the freedom and flexibility to go where they wish. The user interface is colorful yet minimalistic with playful popping noises used as the button click sound effect.

4.1 Creating a ciceroni

Users create a tour by filling in text fields for a title and description, selecting categories from a drop down menu, and choosing public or private settings by selecting a radio button. Touching a text field opens the default iOS keyboard and touching the category drop down menu opens the default iOS selection dialog. See Figure 1 for an example of a blank ciceroni form.

The image shows a mobile application interface for editing a location. At the top, there is a status bar with 'Verizon', signal strength, time '11:56 AM', and battery level '91%'. A red 'Cancel' button is positioned below the status bar. The form consists of several sections: a 'Title' section with a text input field containing the placeholder 'Press here to enter a title'; a 'Description' section with a larger text input field containing the placeholder 'Press here to enter a description'; a 'Categories' section with a horizontal scrollable list showing '0 Items' and a plus icon; and a 'Privacy' section with two radio button options, 'Public' (which is selected) and 'Private'.

Verizon 11:56 AM 91%
Cancel

Title
Press here to enter a title

Description
Press here to enter a description

Categories
0 Items ...

Privacy
☒ Public
☐ Private

Figure 1: Editing a ciceroni

The creation of a location entails choosing a title, filling out the description, choosing a GPS location by making a selection from the map, and taking a picture of the location. See Figure 2 for an example of a blank location form.

Edit Location

Cancel

Title
Press here to enter a title

Description
Press here to enter a description

GPS Location

Lat None

Long None

Set GPS Location

Picture
None

Camera

OR

Enter URL here

Web Links

Add New +

No links

Figure 2: Editing a location

The map screen allows users to set the location by searching for known locations in the search box and then touching anywhere on the map to set a marker. See Figure 3 for an example of the map screen used to select a location.

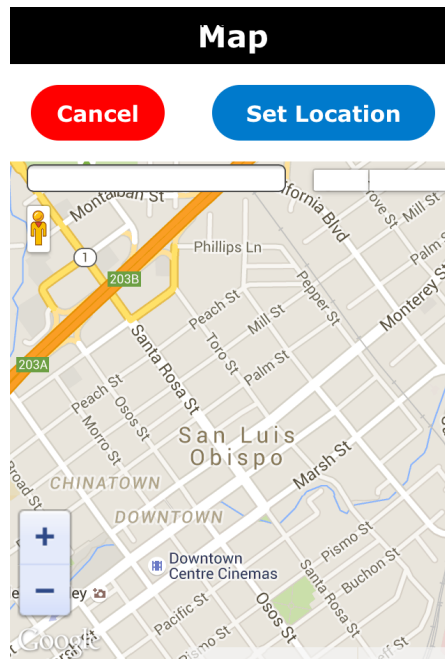
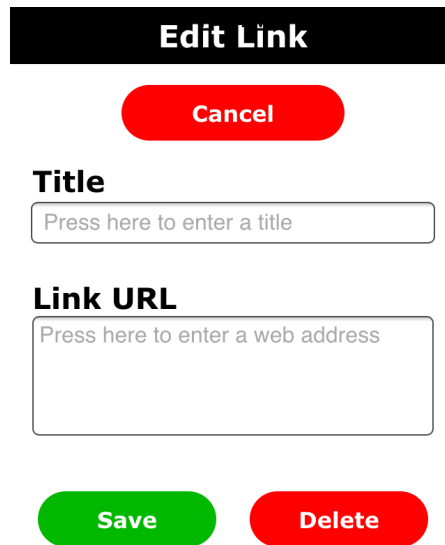


Figure 3: Selecting a GPS location on the map

Some users may choose to include an internet link for a location. Since the entry of the URL is text based, users are expected to copy and paste a URL from the web browser on their device or enter it from memory. See Figure 4 for an example of a blank internet link form.

The form is titled "Edit Link" in a black header bar. Below the header is a red "Cancel" button. The form contains two input fields: "Title" with a placeholder "Press here to enter a title" and "Link URL" with a placeholder "Press here to enter a web address". At the bottom are two buttons: a green "Save" button and a red "Delete" button.

Edit Link

Cancel

Title

Press here to enter a title

Link URL

Press here to enter a web address

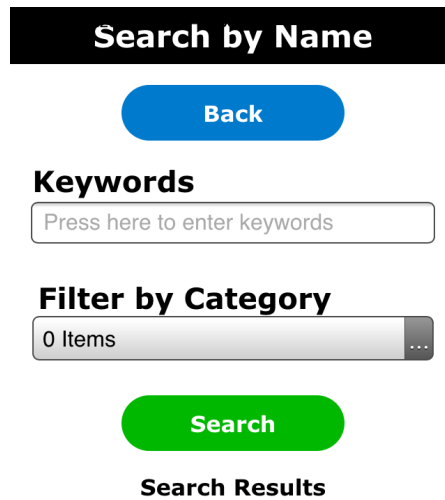
Save **Delete**

Figure 4: Editing a link

Users are free to cancel the addition of a location or ciceroni at any time by touching the red cancel button at the bottom of each page. If users are happy with the information they've entered, they can save it by touching the green save button at the bottom of the page. The entire ciceroni including the information for all locations are saved to the database when clicking save on the ciceroni form.

4.2 Viewing ciceroni

Ciceroni created by other users can be found by searching with a keyword or by proximity to a chosen location. Keyword searches query the database for any ciceroni containing any of the keywords in their title. See Figure 5 for an example of searching by name.



The image shows a mobile application interface for searching by name. At the top is a black header with the text "Search by Name" in white. Below the header is a blue rounded button labeled "Back". Underneath is the section "Keywords" with a text input field containing the placeholder "Press here to enter keywords". Below that is the section "Filter by Category" with a grey rounded button labeled "0 Items" and a three-dot menu icon. At the bottom of the search section is a green rounded button labeled "Search". Below the search section is the text "Search Results".

Figure 5: Searching for ciceroni by name

Ciceroni saved in the database are assigned a location value by averaging the latitude and longitude values of all their locations. Proximity searches query the database for any ciceroni whose location value is within the selected range of the chosen location, whether that be the device's current location or some other specific location chosen on the map. Searches can be further refined by filtering by category. See Figure 6 for an example of searching by proximity.

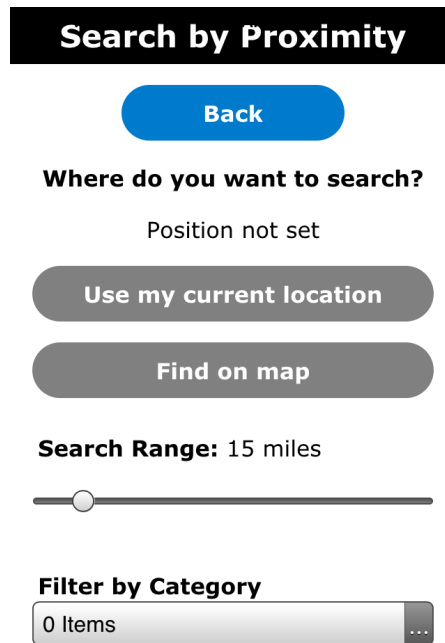


Figure 6: Searching for ciceroni by proximity

When viewing a ciceroni, the title and description are displayed on the first screen. Each location in the list that follows can be touched to open the individual location information. Locations can be viewed on the map all together, or individually. When viewing a location or locations on the map, directions to the selected location can be obtained by clicking the green "Get Directions" button on the map page. Detailed step-by-step directions are also available by touching the orange "View Text Directions" button at the bottom of the map page. See Figure 7 for an example of the map page displaying all locations, directions to a location, and text directions.

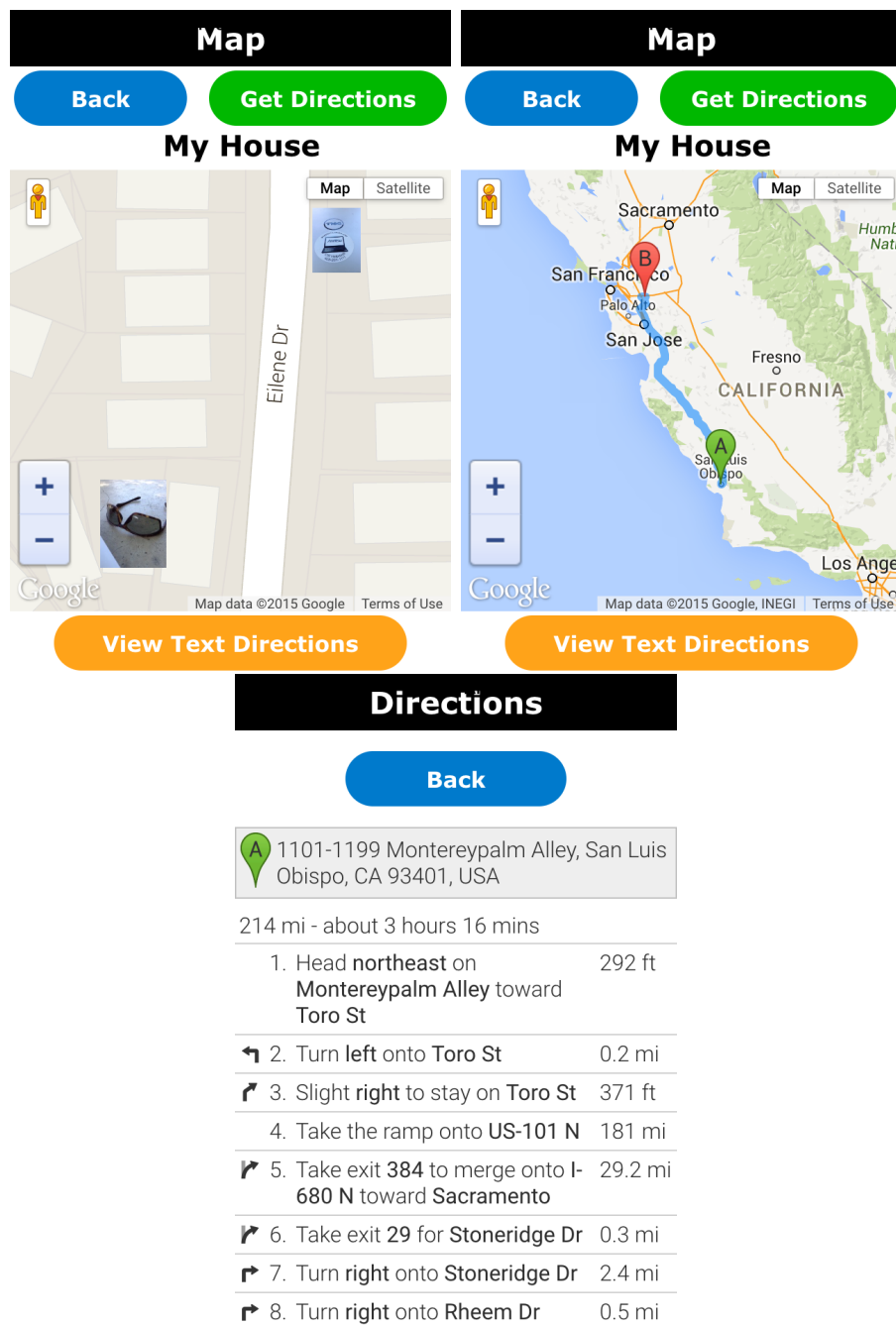


Figure 7: Getting directions to a location

5 Intel XDK Review

The Ciceroni app was developed using the Intel cross-platform development kit (XDK) [10]. A quick review of this relatively obscure IDE is provided here in the hopes that it will be useful for developers in making a decision on which development environment to use for their future work.

5.1 Advantages

The Intel XDK delivers on its main promise of providing a cross platform development environment. It allows developers to write one code set and compile apps on some major mobile operating systems including iOS and Android [10]. It also supports Windows 8 application builds. Other build types that have been deprecated but are still available include Windows Phone, Tizen, Amazon, and Nook [10]. It also allows developers to compile their code as a web app for Chrome, Firefox, or Facebook [10]. Intel XDK comes packaged with the Brackets code editor which is simple, yet sufficient [10]. The XDK enables developers to easily install supported plugins by simply checking a checkbox [10]. It also comes packaged with an emulator that simulates the app on a variety of supported devices for quick testing [10]. In addition to emulation, apps can be debugged on an actual device and configured to produce console output messages [10]. Furthermore, it supports profiling on Android devices, but this feature was not used during the development of Ciceroni [10].

5.2 Disadvantages

Despite its many benefits, the Intel XDK is not perfect. During the course of development of the Ciceroni app, the XDK crashed countless times albeit with no loss of data. Often these crashes occurred while attempting to debug on an iOS device. Debugging output sometimes produced cryptic error messages upon initialization that seemed to have no effect on the app's functionality and error messages produced while building the app were unhelpful in tracking down the cause. The built-in emulator doesn't support certain plugins, specifically the Facebook Connect plugin used in the Ciceroni app, which can make debugging difficult after adding certain features.

6 Future Work

Two features that were originally intended to be included in the final version were not ready for release. Locations would ideally include an audio recording feature which would allow viewers of a cicerone to listen to audio descriptions of each location. This feature was not included because of technical issues with converting audio data to a format supported by the Parse Core database. Ciceroni would ideally include a five heart rating system which would allow users to rate the ciceroni created by others and filter by rating when searching. There was no technical limitation preventing this feature. It was not included because of time constraints and other features received a higher priority. Future work should definitely involve including a rating system, but the audio recording feature is less essential as the app already provides textual descriptions.

7 Conclusions

The Ciceroni app successfully solves the problems it was designed to address. It is simple and minimalistic, yet completely functional. Its main attraction lies in the creative potential that it provides to any user with a Facebook account. Ciceroni aren't limited to be only tours in the traditional sense. For example, they could also contain suggested locations for a day trip or points of interest at a national park.

7.1 Limitations

One limitation of the app is that it forces users to login with Facebook which may slightly impact the number of potential users. Additionally, the Parse Core database is setup with a free account which is limited to 30 requests per second, 20GB of image storage, and 20GB of data storage. However, this limitation is flexible because additional bandwidth and storage can be purchased in the future if necessary.

7.2 Retrospective

Looking back on the development of the Ciceroni app reveals some improvements that could be made to the process. Firstly, the Intel XDK should be avoided. The issues mentioned in Section 5.2 caused delays and significantly extended development time. Coding the app in JavaScript didn't cause any issues or create limitations. The Ciceroni app could have been developed with the majority of the existing code but without the Intel XDK development environment. However a significant amount of time and effort would have to go towards installing and configuring all the Cordova plugins, which was something the XDK did automatically. The responsiveness of the application could be improved by writing the code in Objective C using the Xcode development environment. This is due to native iPhone apps being inherently more efficient than native web apps.

References

- [1] *Stray Boots*. Stray Boots Inc, New York, NY, 2015; software available at <https://www.strayboots.com/>
- [2] Meyer, C., and Harvey, S. *TrackMyTour*. Red House Consulting GmbH and Demobia GmbH. Switzerland, 2015; software available at <http://trackmytour.com/>
- [3] *cordova-plugin-camera*. The Apache Software Foundation, 2012; software available at <https://github.com/apache/cordova-plugin-camera>
- [4] *cordova-plugin-geolocation*. The Apache Software Foundation, 2012; software available at <https://github.com/apache/cordova-plugin-geolocation>
- [5] *cordova-plugin-inappbrowser*. The Apache Software Foundation, 2012; software available at <https://github.com/apache/cordova-plugin-inappbrowser>
- [6] *cordova-plugin-media*. The Apache Software Foundation, 2012; software available at <https://github.com/apache/cordova-plugin-media>
- [7] *phonegap-facebook-plugin*. Nitobi Software Inc, 2010; software available at <https://github.com/Wizcorp/phonegap-facebook-plugin>

- [8] *Parse Core*. Menlo Park, CA. software available at <https://parse.com/products/core>
- [9] *Google Maps JavaScript API*. May 28, 2015; software available at <https://developers.google.com/maps/documentation/javascript/>
- [10] *Intel XDK*. Intel Corporation, 2015; software available at <https://software.intel.com/en-us/intel-xdk>